1. What are the problems in imbalanced data sets and how would you resolve them?

What is an imbalanced dataset?  
An imbalanced dataset is when there are many more instances of a single class label than the other class labels in a dataset. For example, if the class labels are binary and the dataset is 90% 0 and 10%.

Problems of imbalanced dataset and how would you resolve them?

* An imbalanced dataset has two core problems. Firstly, commonly used metrics such as accuracy do not mean much when you have an imbalanced dataset. Therefore, instead of accuracy I would use precision, recall or ROC curves to evaluate the model.
* Furthermore, the classifier may overpredict the majority class. To remedy this over-sampling of the minority class and under sampling of the majority class can be performed on the training set. The over/under sampling will balance the training set.

Note that the model must still be tested on a validation test that follows the original class distribution.

1. What is the curse of dimensionality and how would you resolve it?

What is the curse of dimensionality?  
As the dimensionality of a dataset increases, the data becomes sparser. The curse of dimensionality is that as data becomes sparser, density and distance measures become more meaningless. Thus, clustering algorithms such as K-means do not perform as well.

How to resolve the curse of dimensionality?

* Firstly, I would perform dimensionality reduction using Principal Component Analysis and Feature selection. Both these techniques avoid the curse of dimensionality by reducing irrelevant features and reducing noise as well as reduce time/space computation requirements.

1. Evaluate missing data methods?

* Firstly, you can ignore/delete instances that are missing
  + However, the issue with this method is that if huge portions of the dataset are missing then there is not enough remaining data to train the model correctly
  + Therefore, if large portions of data are missing imputation methods may be better.
* Secondly, three simple imputation methods that can be used are global constant impute, attribute mean imputation and attribute class mean imputation.
  + These methods also have a core problem which is that they add bias to the data and may create new relationships between attributes and the target class that do not actually exist.
* Lastly, what you can do is use the most probable value technique. This technique uses algorithms such as k-nearest neighbours to predict the missing values.
  + This method has problems if there is not enough data to predict the missing values. Further inference algorithms may have other issues such as being trained on noisy data creating incorrect values, producing new relationships between attributes and the class as well as adding bias to the data.

1. How would you adapt an SVM to multi-class classification? Name 2 approaches and explain them.

The two approaches to adapt an SVM to multi-class classification problems are 1 vs rest and 1 vs 1. In both methods we

* 1 vs rest method: This method predicts a single label and aggregates the other labels into the other group. Then binary classification is run. This method can be run across all labels to complete a multi-class classification.
* 1 vs 1 method: This method predicts two labels out of a set of labels and iterates through the set until all pairs of labels were predicted against one another. Then the predictions are combined to create the final output.

1. Explain how Naive Bayes uses the Bayes theorem for classification.

Naïve Bayes

1. Your goal is to build an agent that performs well at chess. Explain how you would approach this.

Answer question using the following structure: experience, target function, target representation, the learning algorithm, the measure of performance and the input encoding.  
  
Agent: A chess player   
Actions: moving chess pieces  
Environment: chess board  
Experience: playing against another chess software action OR database of human chess games  
Problem: find the optimal policy that maximises the chance of winning chess games.  
Learning algorithm: Q-learning

Initially, the agent would have a random policy where chess pieces are moved randomly. Using Q-learning, the Q-matrix can be iteratively updated across multiple simulations of chess games. When the Q-matrix has converged, the policy arrived at (by selecting the action with the maximum Q-value from the current chessboard state) will hopefully make the agent good at chess when playing against a real player.

1. Briefly explain the k-means approach.

K-means is an unsupervised partitional clustering algorithm where the number of clusters (k) must be specified. K-means initially sets random centroids. Then each point is clustered into the closest centroid. The centroids are then re-calculated, and points are once more re-clustered. This process continues until the centroids do not move much anymore and or very few points change clusters.

1. Briefly explain agglomerative clustering approach.

Agglomerative clustering is a hierarchical clustering approach. In this method the points start as individual clusters. Then the closest pair of clusters are merged together. The stop criterion for this algorithm are when only one/k clusters are left.

1. Given the following scenario would you chose to use k-means or agglomerative clustering. You have a dataset of 1000 data points and want to have a **stable** clustering. You want to also want to know what the number of clusters should be after the clustering has completed.

I would choose agglomerative clustering. This is because in k-means we have to know how many clusters we want before hand and in agglomerative clustering reveals the number of clusters there should be by examining the tree structure and cutting the dendrogram in the desired place to generate clusters.  
Furthermore, agglomerative clustering also creates stable clustering because it is not sensitive to seeding the start points like K-means.

1. You have a discussion with a friend, and your friend suggests that you can capture virtual drift by monitoring the error rates within the model. Is this argument true? Explain why or why not.

This is not true. Virtual drift is where new information in the stream changes the distribution in the feature space but does not cause a change in the classification decision boundary. Therefore there is no degradation in accuracy/error rates and therefore monitoring error rates will not capture virtual drift.

1. Explain why it is necessary to have a drift detector for data stream analysis. Explain one drift detector as well.

In stream analysis, if the distribution of the new incoming data is different from the data the model was trained on then the model will perform poorly. This is called concept drift and drift detectors are able to detect and adapt models to concept drift – this is why are they are necessary in stream analysis.

One example of a drift detector is cumulative sum (CUSUM) techniques such as Page Hinkley. In this technique the drift detector signals concept drift has occurred when the mean of the error rate of the input data differs greatly from 0.

1. Explain what the LOF anomaly detection is and its advantages.

LOF (Local Outlier Factor) anomaly detection is a density-based proximity detection method for detecting outliers in a dataset. In this method for each datapoint the density of the local neighbourhood is computed and the LOF score for each point is computed. The LOF score for each point is the average of the ratios of the density of the point compared with the density of its nearest neighbours. Points with large LOF scores are more likely to be classified as outliers.

Advantages:

* Can handle clusters of varying densities to find outliers. This is because it uses relative densities.
* LOF can be used for data with no labels

1. The ’database’ below has four transactions. What are the frequent itemsets, and association rules that can be found in this set, if the minimum support is 60% and the minimum confidence is 80%? Explain your working using Apriori approach.

Trans\_id Itemlist

T1 {K, A, D, B}   
T2 {D, A C, E, B}   
T3 {C, A, B, E}   
T4 {B, A, D}

* Generate

1. Explain the impact of minimum support threshold on the number of frequent itemsets that may be found.

As the threshold increases, the number of frequent itemsets that maybe found becomes smaller. This decreases the runtime of the algorithm.